Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) An authentication engine architecture for ~~an~~ a multi-loop, multi-round authentication algorithm, comprising:

a first instantiation of a multi-round authentication algorithm hash round logic in an inner hash engine;

a second instantiation of a multi-round authentication algorithm hash round logic in an outer hash engine;

a dual-frame payload data input buffer configured for loading one new data block while another data block ~~one~~ is being processed in the inner hash engine;

an initial hash state input buffer configuration for loading initial hash states to the inner and outer hash engines for concurrent inner hash and outer hash operations; and

a dual-ported ROM configured for concurrent constant lookups for both inner and outer hash engines.

2. (Original) The authentication engine architecture of claim 1, wherein the multi-loop, multi-round authentication algorithm is HMAC-MD5.

3. (Original) The authentication engine architecture of claim 1, wherein the multi-loop, multi-round authentication algorithm is HMAC-SHA1.

4. (Original) The authentication engine architecture of claim 1, wherein at least one of the inner and outer hash engines is configured to implement hash round logic including at least one addition module comprising:

a plurality of carry save adders for computation of partial products; and a carry look-ahead adder for computation and propagation of a final sum.

5. (Original) The authentication engine of claim 4, wherein the carry save adders and the carry look-ahead adder are configured such that addition computations are conducted in parallel with round operations.

6. (Original) The authentication engine architecture of claim 3, wherein at least one of the inner and outer hash engines is configured to implement hash round logic comprising:

five hash state registers;

one critical and four non-critical data paths associated with the five registers, such that in successive SHA1 rounds, registers having the critical path are alternative.

7. (Original) The authentication engine architecture of claim 6, wherein said hash round logic is implemented such that eighty rounds of an SHA1 loop are collapsed into forty rounds.

8. (Original) The authentication engine architecture

of claim 3, wherein at least one of the inner and outer hash engines is configured to implement hash round logic comprising:

five hash state registers;

a 5-bit circular shifter;

an add5to1 adder module having a plurality of CSAs and a CLA adder;

a 30-bit circular shifter; and

an add4to1 adder module having a plurality of CSAs and a CLA adder.

9. (Original) An authentication engine architecture for a multi-round authentication algorithm, comprising:

a hash engine configured to implement hash round logic for a multi-round authentication algorithm, said hash round logic implementation including at least one addition module comprising,

a plurality of carry save adders for computation of partial products, and

a carry look-ahead adder for computation and propagation of a final sum.

10. (Original) The authentication engine of claim 9, wherein the carry save adders and the carry look-ahead adder are configured such that addition computations are conducted in parallel with round operations.

11. (Original) The authentication engine architecture of claim 9, wherein the multi-round authentication algorithm is

MD5.

12. (Original) The authentication engine architecture of claim 9, wherein the multi-round authentication algorithm is SHA1.

13. (Original) The authentication engine architecture of claim 12, wherein the hash round logic implementation comprises:

five hash state registers;

a 5-bit circular shifter;

an add5to1 adder module having a plurality of CSAs and a CLA adder;

a 30-bit circular shifter; and

an add4to1 adder module having a plurality of CSAs and a CLA adder.

14. (Original) An authentication engine architecture for an SHA1 authentication algorithm, comprising:

at least one hash engine configured to implement hash round logic comprising:

five hash state registers;

one critical and four non-critical data paths associated with the five registers, such that in successive SHA1 rounds, registers having the critical path are alternative.

15. (Original) The authentication engine architecture of claim 14, wherein said hash round logic is implemented such

that eighty rounds of an SHAI loop are collapsed into forty rounds.

16. (Original) A method of authenticating data transmitted over a computer network, comprising:

receiving a data packet stream;

splitting the packet data stream into fixed-size data blocks; and

processing the fixed-size data blocks using a multi-loop, multi-round authentication engine architecture having a hash engine core comprising an inner hash engine and an outer hash engine, said architecture configured to,

pipeline hash operations of said inner hash and outer hash engines,

collapse and rearrange multi-round logic to reduce rounds of hash operations, and

implement multi-round logic to schedule addition computations to be conducted in parallel with round operations.

17. (Original) The method of claim 16, wherein said pipelining comprises performance of an outer hash operation for one data payload in parallel with an inner hash operation of a second data payload in a packet stream fed to the authentication engine.

18. (Original) The method of claim 17, wherein a dual-frame input buffer is used for the inner hash engine.

19. (Original) The method of claim 18, wherein initial hash states for the hash operations are double buffered for concurrent inner hash and outer hash operations.

20. (Original) The method of claim 19, wherein concurrent constant lookups are performed from a dual-ported ROM by both inner and outer hash engines.

21. (Original) The method of claim 16, wherein the multi-loop, multi-round authentication algorithm is MD5.

22. (Original) The method of claim 16, wherein the multi-loop, multi-round authentication algorithm is SHA1.

23. (Original) The method of claim 22 wherein said scheduling of additions comprises:

conducting a 5-bit circular shift on data from a first register;

adding an initial hash state in a second register, a first payload data block, a first constant, and the result of a function (F_t) of the initial hash states in third, fourth and fifth additional registers with an add5to1 adder module having a plurality of CSAs and a CLA adder;

conducting a 30-bit circular shift on data from the third additional register; and

adding the initial hash state in the fourth additional register to a second payload block, a second constant, and the result of a function (F_t) of the initial hash states in the first and fifth registers and the shifted hash state of the third register with an add4to1 adder module having a plurality of CSAs and a CLA adder.

24. (Original) The method of claim 22, wherein said collapsing and rearranging of the multi-round logic comprises:

providing five hash state registers; and

providing data paths from said five state registers such that four of the five data paths from the registers in any SHA1 round are not timing critical.

25. (Original) The method of claim 24, wherein, in successive SHA1 rounds, registers having the critical path are alternative.

26. (Original) The method of claim 25, wherein eighty rounds of an SHA1 loop are collapsed into forty rounds.

27. (Original) A method of authenticating data transmitted over a computer network, comprising:

receiving a data packet stream;

splitting the packet data stream into fixed-size data blocks; and

processing the fixed-size data blocks using a multi-round authentication engine architecture, said architecture implementing hash round logic for a multi-round authentication algorithm configured to schedule addition computations to be conducted in parallel with round operations.

28. (Original) The method of claim 27 wherein said hash round logic comprises:

conducting a 5-bit circular shift on data from a first register;

adding an initial hash state in a second register, a first payload data block, a first constant, and the result of a function (F_{t}) of the initial hash states in third, fourth and fifth additional registers with an add5to1 adder module having a plurality of CSAs and a CLA adder;

conducting a 30-bit circular shift on data from the third additional register; and

adding the initial hash state in the fourth additional register to a second payload block, a second constant, and the result of a function (F_t) of the initial hash states in the first and fifth registers and the shifted hash state of the third register with an add4to1 adder module having a plurality of CSAs and a CLA adder.

29. (Original) A method of authenticating data transmitted over a computer network using an SHA1 authentication algorithm, comprising:

providing five hash state registers; and

providing data paths from said five state registers such that four of the five data paths from the registers in any SHA1 round are not timing critical.

30. (Original) The method of claim 29, wherein, in successive SHA1 rounds, registers having the critical path are alternative.

31. (Original) The method of claim 30, wherein eighty rounds of an SHA1 loop are collapsed into forty rounds.